一种面向服务的数据集成平台①

房立芳 杨永超 (中国科学技术大学 自动化系 安徽 合肥 230027)

摘 要: 提出一种松散耦合的数据集成解决方案，将异构数据源作为服务提供者从集成系统中分离出来，采用目前较为流行的基于中介器/包装器的数据集成方法，自底向上地将数据源集成为统一的数据视图——集成处理包，并初步实现了一个原型系统。重点介绍了该系统的体系构架，以及数据集成处理包的结构设计和创建过程，最后，通过实例验证了此方案的灵活性和有效性。

关键词: 数据集成; 中介器/包装器; 数据集成处理包

Service-Oriented Data Integration Platform

FANG Li-Fang, YANG Yong-Chao (Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: This paper presents a loosely coupled data integration solution. It separates the heterogeneous data sources from the data integration system as a service provider, and integrates the heterogeneous data sources into a unified data view called data integration processing package by the bottom-up way using the popular mediator/wrapper approach, and implements a prototype system initially. This paper focuses on the architecture of the system, as well as the structure design and the creation process of the data integration processing package, Finally, verifies the flexibility and validity of this solutions by experiment.

Keywords: data integration; mediator / wrapper; data integration processing package

1 引言

随着信息化的不断推进，“信息孤岛”现象日益严重。采用数据集成的方法解决异构数据源的语法/语义异构，为用户提供统一的查询接口受到了越来越广泛的重视[1-3]。数据集成任务包括两部分：一是将分布异构的数据源集成为统一的数据视图；二是处理用户的查询请求。在传统数据集成中，这两部分是紧密耦合在一起的，数据源被动接受用户层的查询请求，导致用户的查询负担过重。

针对上述问题，本文借鉴 Web Service 思想[4]，其体系构架如图 1 所示，将数据源作为服务提供者从集成系统中分离出来，以数据源为研究中心，自底向上地将部分数据源或数据源中的部分数据子集提供给上层用户，以减少用户的查询负担。本文采用目前比较流行的基于中介器/包装器的数据集成方法[5]，将异构数据源集成过程描述为数据集成处理包，进而封装为数据服务单元的形式对外提供服务。这种面向服务的数据集成解决方案，可以使用户避免直接访问源数据，从而大大减少了用户的信息搜索负担。

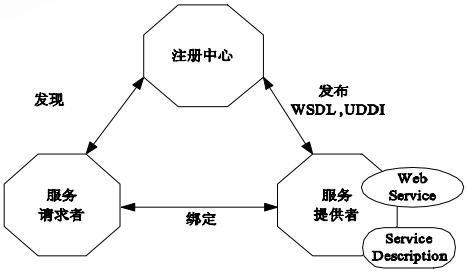

图 1 Web Service 体系构架

2 系统结构

面向服务的数据集成平台(Service-Oriented

① 收稿时间:2010-03-28;收到修改稿时间:2010-04-27

Data Integration Platform，SODIP)是基于数据服务匹配的数据集成系统(Data Integration System based on Data Service Matching, DSM-DIS)[6]的重要组成部分(鉴于篇幅原因，对 DSM-DIS 系统不做过多介绍，其体系结构如图 2 所示)。作为数据源层与服务匹配层的连接桥梁，SODIP 系统以服务提供者的角色，自下而上对遗留系统进行了数据的抽取、合成、传递、输出等操作，抽象出核心服务，最终完成数据服务单元 DS-CELL(Data Service Cell)的构建。在系统实现方面，SODIP 采用 MVC(Model-View-Controller，模型-视图-控制器)设计模式，利用分层架构，降低了系统的耦合性。

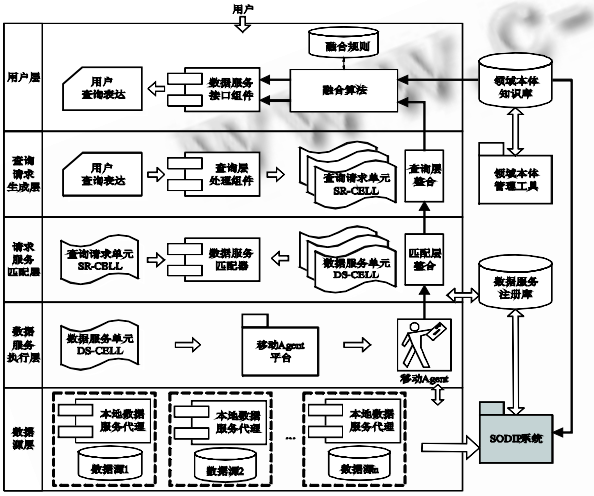

图 2　DSM-DIS 系统体系结构

2.1 视图层

视图层是人机交互界面。视图层包含各种可视化视图控件，如窗口容器、面板、按钮、文本框、用户操作监听器等，收集服务提供者提供的请求参数，传递给控制层，并接受控制层返回的信息，更新视图。

2.2 控制层

控制层用于协调视图层和模型层的操作。SODIP 系统有一个主控制器(MainController)充当中介，使各个子控制器之间可以松散耦合，只需关心和主控制器的交互，使多对多的关系变成了一对多的关系，可以降低系统的复杂性，提高可扩展性。

2.3 模型层

模型层即各个组件的实体模型，即 java 实体对象。它对外提供各种处理接口，处理用户请求，并将结果返回给控制层。模型层主要包括两个大类：源组件 SourceComponent 和任务组件 TaskComponent。

SourceComponent 包括待集成数据源和目的数据源，可以是数据库或者格式化的文本文档。如果是数据库则还有连接 URL、用户名、密码、数据库的类型、驱动信息等属性；如果是格式化的文档，则没有类型和驱动属性。

TaskComponent 包括数据抽取、数据合成、数据传递等三个组件，每个组件都有一个执行顺序属性(ExecuteOrder)，表示系统在执行数据集成处理包时各个任务的执行次序，另外每个组件都有执行语句、结果表结构等属性。

2.4 数据访问层

数据源可能是典型的关系型数据库系统，也可能是 Web 数据、XML 文件、文本文件等半结构化的数据存储系统。不同类型数据源有不同的访问方式，如图 3 所示。

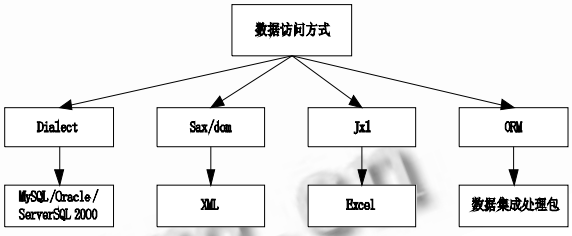

图 3　各种异构源的访问方式

对于各种关系型数据源，本文使用 Hibernate 技术，在数据源层之上添加持久化层，将对异构数据源的操作，统一转化为对 Java 对象的操作，从而避免了各个异构源查询语言的差异对数据源访问带来的不便。我们采用数据库的名称来区分不同数据库的配置文件(数据名.cfg.xml)，使用 Hibernate 的查询语言——HQL 语言，实现对异构数据源的透明访问，避免了为每一个数据源设计单独包装器(Wrapper)。

3　数据集成处理包

数据集成处理包是整个集成系统的核心组件，它从一个高度抽象的层次对数据集成过程进行语义描述，可以“记录”整个集成过程。经编译后的数据集

成处理包生成中间件对象，可以和 JavaScript、CORBA 等对象进行互操作。该集成处理包本质上是一个有向无环图，各个组件按照它们在通路上的位置，从上游到下游依次被执行。

3.1 功能描述

数据源组件：用来表示特定类型的数据源。采用关系模型作为公共模型，其他数据源则需要先转化为关系模型表达的模式，再进行其他操作。

数据抽取组件：用来连接数据源组件和数据合成组件。通过该组件可以查看已连接数据源中所有表及表的结构，然后选择需要的表，将其数据传递到合成组件。

数据合成组件：用来实现多路视图的合成。一个数据合成组件可以连接多个数据抽取组件，其输出可以通过传递组件传递到下游合成组件或目的库组件。该组件提供了灵活、多样的合成方法，例如，多表连接，字段名修改、过滤、分组、排序等操作，进行不同字段值、不同数据类型或不同值度量单位等变换，解决集成时语义层的异构问题；在最后一级合成中间表形成后，可以通过一对一映射操作，将其映射为需要的 XML 格式，解决结构层异构问题。

数据传递组件：用来连接数据合成组件和目的库组件。它可以将合成组件的数据传递给下游合成组件或直接输出到目的库组件中。

目的库组件：用来表示特定的目的库组件，是最终查询结果的存储容器。

3.2 处理包结构设计

```
<?xml version="1.0" encoding="UTF-8"?>
<package>
<dataSource id=" ">
<databaseName type="Mysql" url=" " username=" " password=" " ></databaseName>
</dataSource>
<dataExtraction id=" "> …… </dataExtraction>
<dataSyn id=" ">
<input>……</input>
<queryString></queryString>
<result_table xposition=" " yposition=" " width=" " height=" "></result_table>
……
<maptable>
<row>……</row>
</maptable>
</dataSyn>
<dataTransfer id=" ">……</dataTransfer >
<dataout id=" ">……</dataout>
</package>
```

说明：<dataSource/>,<dataExtraction/>,<dataSynList/>，<dataTransfer/>,<dataout/>分别对应于数据源组件、数据抽取组件、数据合成组件、数据传递组件和数据输出组件。数据抽取组件上端只能连接数据源，下端连接合成或输出组件；数据合成组件只能连接数据抽取或数据传递组件；数据传递组件上端只能连接合成组件，下端连接合成或输出组件。在满足以上的约束条件的前提下，以上的标签可以按操作次序多次出现。

3.3 处理包的创建过程

(1) 用户发送新建包请求。

(2) 配置数据源。选择要操作的表所在的数据源，定义其连接属性。

(3) 定义数据抽取组件。在本平台中，规定每个数据抽取组件上端只能连接一个数据源。在该环节中可对特定的数据源进行查看、过滤、转换等操作。

(4) 定义数据合成组件，实现视图合成。合成组件上端可以连接数据抽取组件或数据传递组件，下端只能连接数据输出组件。在合成环节可以进行灵活多样的合成操作，一个集成过程中可以有零个或多个合成组件，按其在通道上的位置依次被执行。

(5) 定义输出及目的库组件。将合成的结果保存在选定的目的库中。

(6) 保存数据处理包到数据库或者 XML 文件中。

(7) 单步调试/测试。

3.4 数据服务单元

基于SODIP平台创建的数据集成处理包会被封装为一种基于 OWL-S 的服务请求单元(DS-CELL)。OWL-S[7]是 Darpa 组织推出的新一代语义 Web 服务描述框架，它是一种用来描述 Web 服务的属性和功能的 OWL 本体规范，旨在支持语义 Web 服务的自动组

合和调用，帮助用户和代理查询、发现、调用、组合和监控语义 Web 服务，试图利用语义描述和逻辑推理实现服务匹配的自动化和智能化。OWL-S 主要包括三部分：ServiceProfile、ServiceModel 以及 Service Grounding。参考 OWL-S 的本体服务描述模型，我们设计的数据服务单元 DS-CELL 主要包括服务简介(ServiceProfile)和数据处理(DataProcess)两个部分。ServiceProfile 包含数据服务的基本信息描述以及数据服务的输入、输出概念集，以便与用户的查询请求进行匹配。DataProces 主要包括数据集成处理包的信息，它是对数据集成处理过程的逻辑说明。

4　实例验证

4.1 原型系统主界面

原型系统主界面截图如图 4 所示，可以直接单/双击界面上的图元，进行属性编辑，在任务工作区依次进行数据源配置、数据抽取、数据合成、数据传递和数据输出等操作。

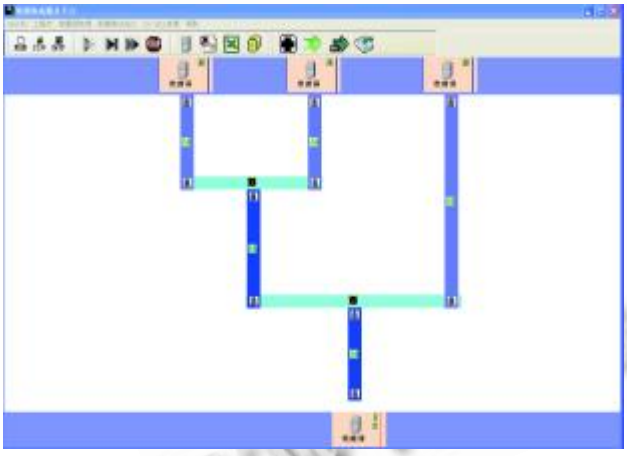

图 4　SODIP 原型系统主界面截图

4.2 异构数据源分析

有两个数据源，数据源 1：在 MySQL 中建立的数据源，数据模式包含关系：student1(ID,NAME,GENDER,COURSE,SCORE)，表示 student1 含有的属性依次为学号、姓名、性别、所选课程和分数，其中性别用“1”表示男，“0”表示女；数据源 2：在 Oracle10i 中建立的数据源，数据模式包含关系 student2(ID,NAME,AGE,DEPT)，表示 student2 含有的属性依次是学号、姓名、年龄、所在系别。而对于某些用户，如公寓管理人员，可能只需了解所有性别为“女”的学生的学号、姓名、所属系别等信息就可以了，所以我们只需从两个的异构数据源中抽取这些有用信息提供给用户。

4.3 测试

对表 student1，表 student2 分别执行完数据源配置和数据抽取操作后，集成工具自动将两个表添加到合成环节，图 5 给出了该平台部分属性窗体实现的屏幕截图。我们可以在合成环节对两表进行连接、字段值修改等操作。当然，也可以根据需要进行一系列其他变换，如过滤、分组、排序等操作，或如图 4 所示将前一级合成的结果，作为下一级的一个输入，继续与其他源合成，实现灵活的多级合成操作。

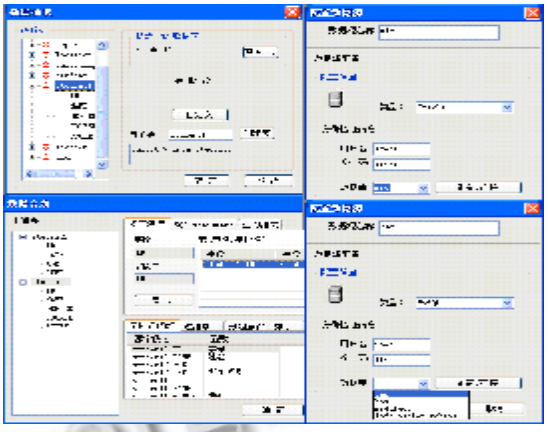

图 5　SODIP 平台部分属性窗体实现的屏幕截图

合成结果如图 6 所示：

图 6　结果显示窗口屏幕截图

5 结束语

本文以友好的图形界面的方式，提出了一种轻量级、松散耦合的数据集成方法。通过引入数据源组件、数据抽取组件、数据合成组件、数据传递组件等抽象描述语言，将数据集成过程描述为抽象的数据集成处理包的形式，表达了复杂的数据集成语义。该平台可以有效屏蔽异构数据源的差异，解决数据集成面临的语法/语义冲突，仅对部分数据源或数据源中的部分数据子集进行集成，减轻用户的查询负担。目前，我们已将此方法应用于实验室大型项目——基于数据服务匹配的数据集成系统中，作为实现该系统在数据源端有关服务组件的基础，实践收到了良好的效果。

参考文献

1 Halevy A, Rajaraman A, Ordille J. Data Integration: The Teenage Years. Proceedings of the 32nd international conference on Very large data bases, 2006:9－16.

2 Lenzerini M. Data Integration: A Theoretical Perspective. Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 2002:233－246.

3 陈跃国,王京春.数据集成综述.计算机科学, 2004,31(5):48－51.

4 Booth D, Haas H, McCabe F, Newcomer E, etal. Web Services Architecture. [2010-01-08]. http://www.w3.org /TR/ws-arch/.

5 Chawathe S, Garcia-Molina H, Hammer J. The TSIMMIS Project: Integration of Heterogeneous Information Sources. Proceedings of the 10th Meeting of the Information Processing Society of Japn, 1994:7－18.

6 谢兴生,张一鸣,余银,等.一种支持智能匹配检索的数据集成系统.模式识别与人工智能,2009,22(1):40－46.

7 David Martin, Mark Burstein, Jerry Hobbs,etc. OWL-S: Semantic Markup for Web Services. [2010-03-09]. http://w